VRIJE
UNIVERSITEIT
BRUSSEL

Analog Electronics lab session

Alican, Guarav, Lucas, Sriram, Thomas

The TA team

Lucas Santana
Master's in Aeronautics Institute of Technology (Brazil, 2019)
Works with delta sigma analog to digital converters

Alican Çaglar
Master's in Istanbul Technical University (Turkey, 2019)
Works with redout circuits for quantum computing

Thomas Gorzka
Master's in RWTH Aachen (Germany, 2019)
Works with transceiver circuits for optical and wireline communication

Sriram Balamurali
Master's in TU Delft (The Netherlands, 2019)
Works with millimeter wave frequency generation

Guarav Agrawal
Master's in IIT Madras (India, 2015)
Works with millimeter wave transceivers for high-speed wireless communications

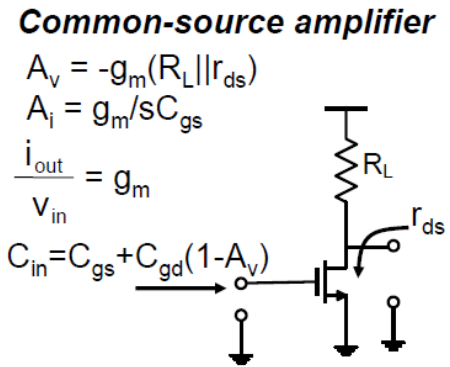
- Session 1: essentials of the MOS transistor
- Session 2: cascade of CS amplifier stages and Miller compensation
- Session 3: design of differential pairs and of an OTA
- Session 4: notions of noise in circuits

Summary

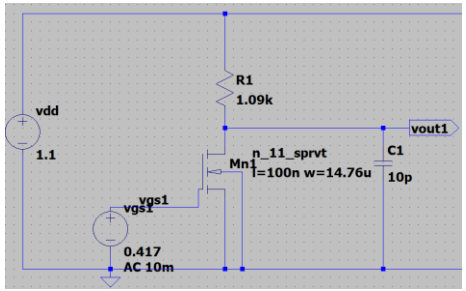
- Session 1:
  - How to use the Matlab and the LTSpice tool
  - MOS design curves and trade-offs
  - Common source amplifier with resistive load
  - Common source amplifier with PMOS load

Core differences between lectures and lab classes

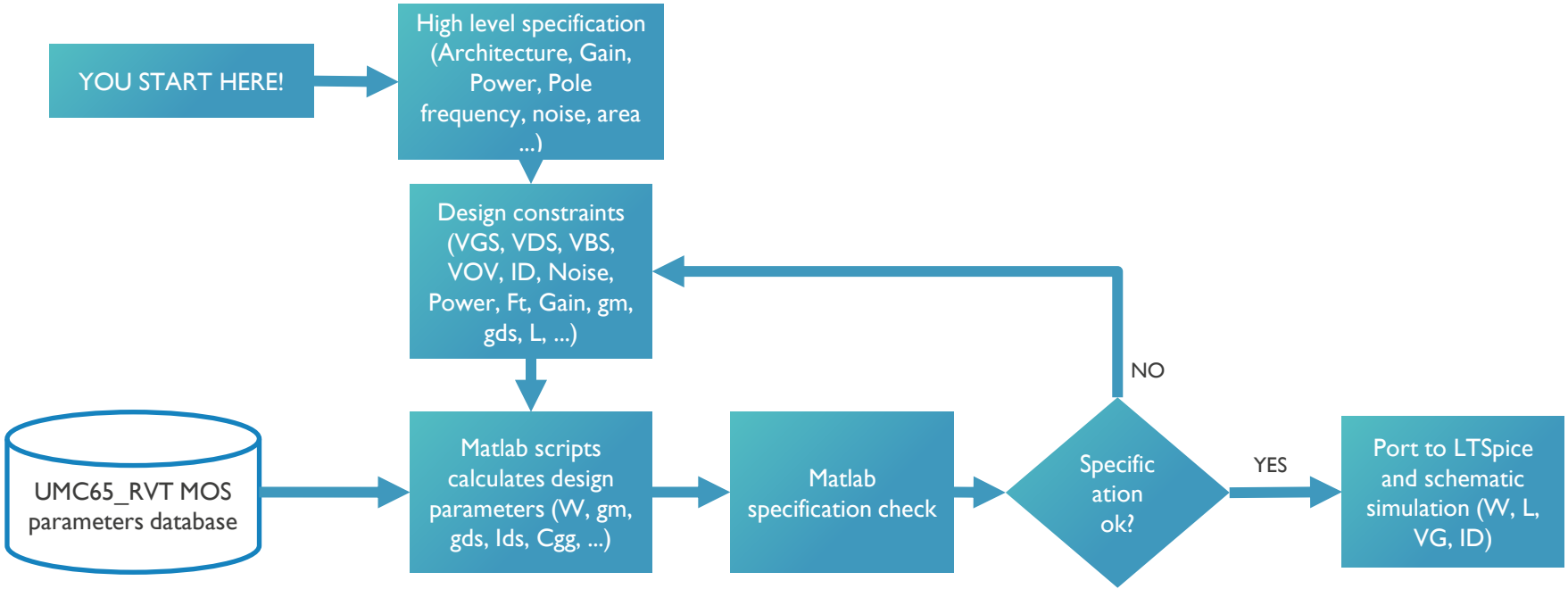
- Lectures:
  - Analysis of given architectures
  - Modeling of transistors and complex circuits
  - Finding equations for performance parameters (gain, bandwidth, power,...)



- Lab:
  - Quantitative design of the known architectures
  - How to size and bias a transistor (W, L, Vgs, Vds) to achieve a certain performance using Matlab
  - How to simulate a circuit using SPICE to validate the correctness of design

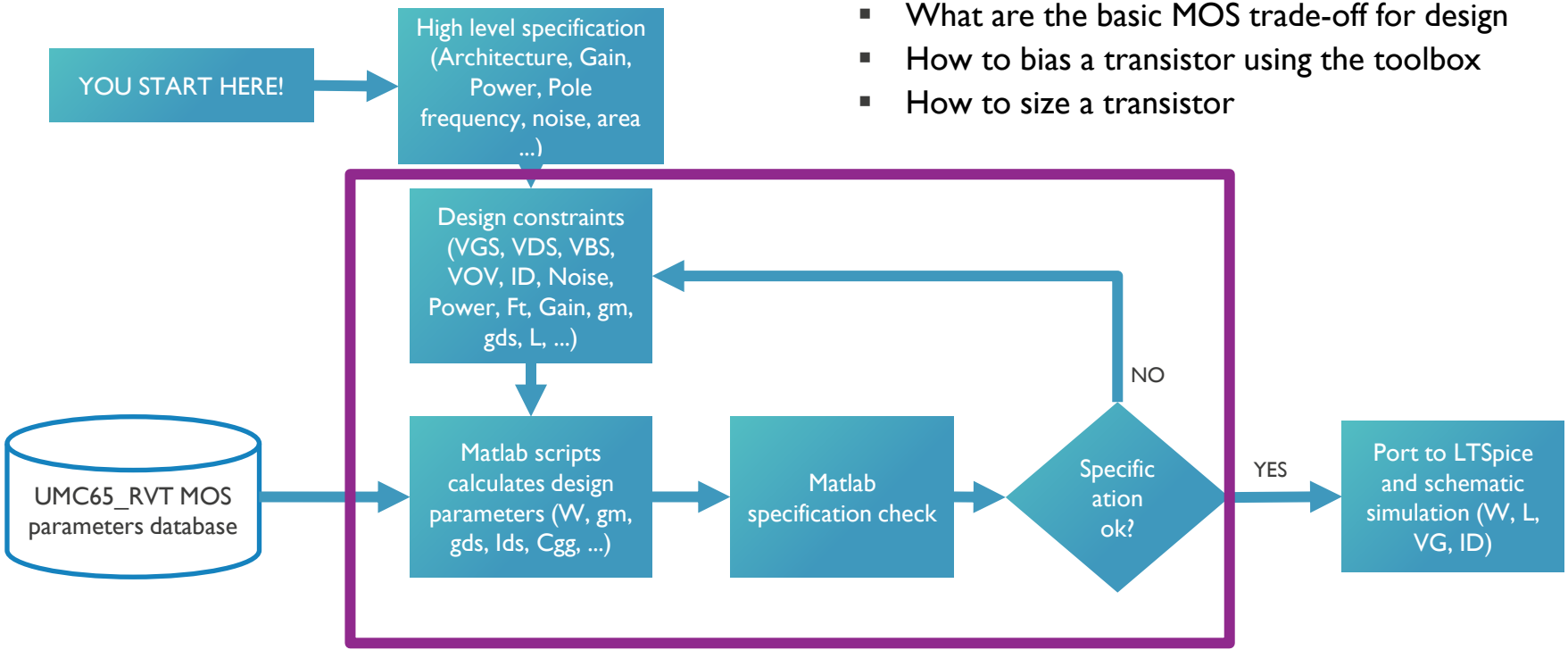


How does the Design process work



How does the design process work

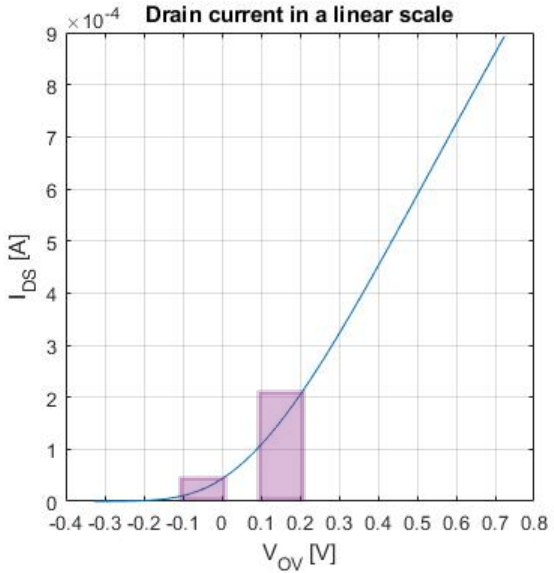
- What do you need to understand after this class?
  - What are the basic MOS trade-off for design
  - How to bias a transistor using the toolbox
  - How to size a transistor

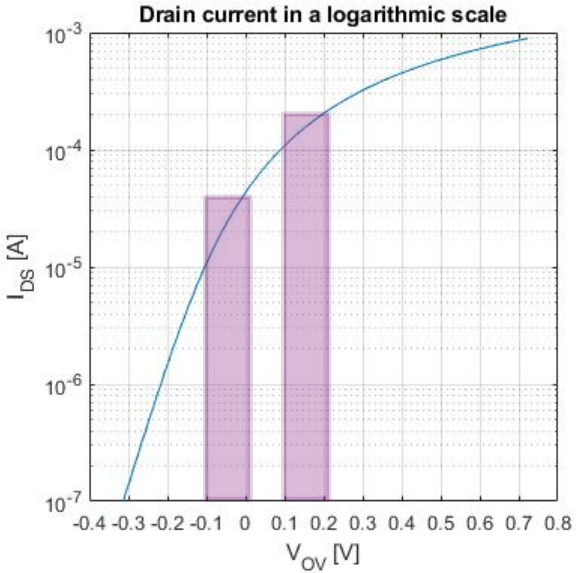


Building some intuition

- Do exercise 1 and exercise 2
  - ETA: 15 minutes

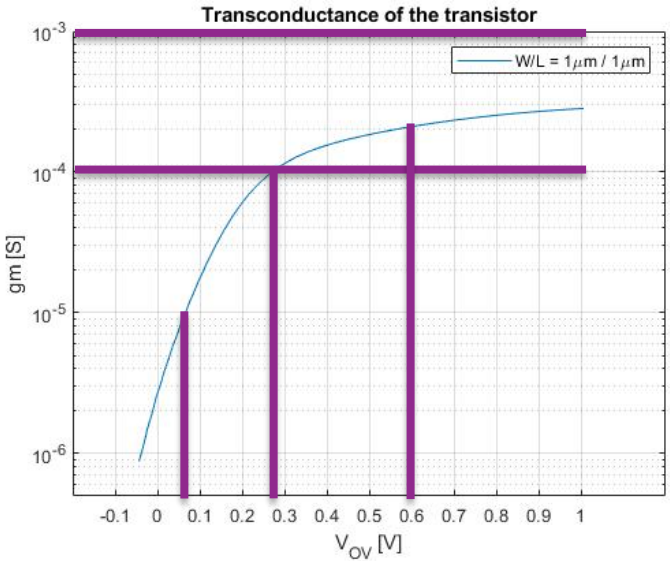
Exercise 1





- WI: vov < -0.1V or **0V**
- MI: -0.1V or **0V**< vov <**0.2V** or 0.3V
- SI: vov > **0.2V** or 0.3V

Exercise 2



- 0.1mS
  - Directly from the graph: vov = 0.3V -> VGS =0.6V
- 1mS – multiple choices
  - Take 10x 0.1mS in parallel with VGS = 0.6V
  - Take 5x 0.2mS in parallel with VGS = 0.9V
  - Take 100 x 0.01mS in parallel with VGS = 0.35V

What does a Matlab code looks like

housekeeping

```
%% Adding paths + Loading MOS tables
addpath(genpath('circuitDesign'));
addpath(genpath('functions'));
addpath(genpath('models'));

w = warning ('off','all'); % Turn off warnings

clear;
close all;
clc;

load ('UMC65_RVT.mat');
```

- Loading functions and .mat file

```
%% Initialize everything
designkitName    = 'umc65';
circuitTitle     = 'Analog Design - Session 1';

%Declaration of the circuit components
elementList.nmos = {'Mn1','Mn2','Mn3','Mn4'};
elementList.pmos = {};

spec.VDD         = 1.1;
choice.maxFingerWidth = 10e-6;
choice.minFingerWidth = 200e-9;
simulator        ='spectre';
simulFile        = 0;
simulSkelFile    = 0;
analog = cirInit('analog', circuitTitle, 'top', elementList, spec , choice,...
    designkitName, NRVT, PRVT, simulator, simulFile, simulSkelFile);

analog           = cirCheckInChoice(analog, choice);
```

- Defining the transistors used in the circuit (Mn is a struct with the transistor parameters as fields)
- Defining technology constraints

Plotting the MOS IDS curves

Session1_part1.m

```
%% Circuit
disp('        Vd            ');
disp('        |             ');
disp('  Vg---Mnx            ');
disp('        |             ');
disp('        Vs            ');

fprintf('\n--- First Exercise: Designing transistor from scratch ---\n');
```

- Fixed parameters: W, L
- Sweep parameters: VGS, VDS
- Calculated parameters: IDS (to be plot), VTH (byproduct of bias point) and other Operating point parameters (gm, gds, ...)

Writing the code together

Useful functions

tableValueWref

```
tableValueWref retrieving the value of an intrinsic MOS parameter for
the reference width, directly from a given table.

   RESULT = mosIntValueWref(PARAM, TABLE, LENGTH, VGS, VDS, VSB) returns
   the value of the MOS operating point parameter PARAM
   (specified as a string) for a given TABLE.
   PARAM must be exist as a field of the given TABLE.
   Possible names for PARAM can be found by running tableDisplay(TABLE).

   EXAMPLE :

      id = tableValueWref('ids', N, 0.18e-6, 0.7, vdd/2, 0)
```

- To extract the VTH value (weak dependency with MOS W)

mosNfingers

```
mosNfingers computation of the number of fingers for a MOS transistor.
```

- To slice the Width according to the maximum desired finger width

mosOpValues

```
mosOpValues computation of all operating parameters of a MOS transistor

   MOS = mosOpValues(MOS) computes the value of intrinsic and extrinsic
   operating point parameters (in S.I. units) of a given transistor MOS. The
   datastructure of the transistor which is specified as an argument is also
   returned. However, after this function has returned, all fields of the
   transistor related to intrinsic and extrinsic operating point parameters
   have been added to the datastructure of the transistor. Existing extrinsic
   geometry parameters are NOT recomputed. Also, the
   values of vgb, vgd and vdb are computed (or updated) from vgs, vds and vsb.

   EXAMPLE :

      Mn1 = mosOpValues(Mn1)
```

- To calculate all the other MOS parameters, in this case it is mainly Mn.ids

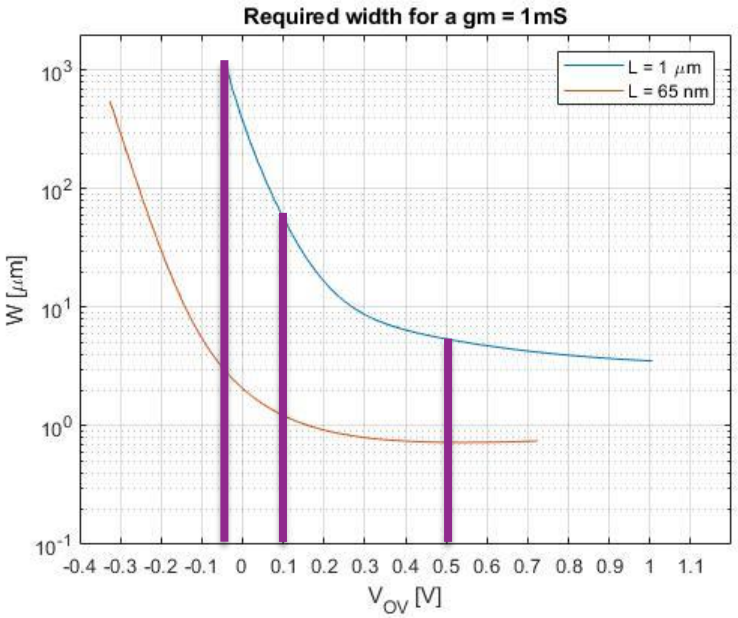
Analysis

- IDS graph
  - Reset to X-Z view: which regions can you see?
  - Reset to Y-Z view: which regions can you see?
- MOS saturation graph
  - Reset to Y-X view: can you discuss about the trends in Vdsat (Vds after which the transistor is in saturation)?

Building some intuition

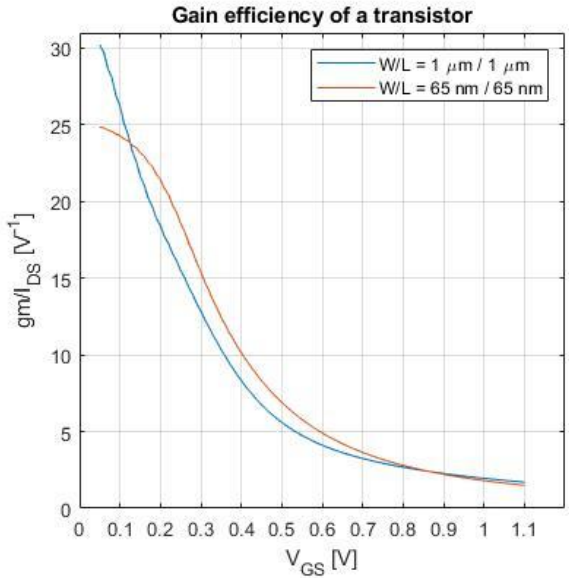
- Exercise 3 and 4
  - ETA: 15 minutes

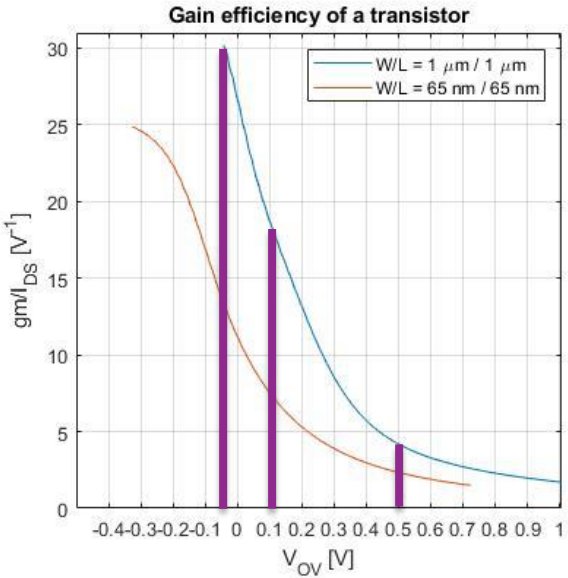
Exercise 3



- Assuming $gm = W \times f(CMOS\ Parameters\ Constant)$
  - $WI: V_{OV} = -0.05V \rightarrow W = 1000\mu m@1mS \rightarrow W = 10000\mu m@10mS$
  - $MI: V_{OV} = 0.1V \rightarrow W = 50\mu m@1mS \rightarrow W = 500\mu m@10mS$
  - $SI: V_{OV} = 0.5V \rightarrow W = 5\mu m@1mS \rightarrow W = 50\mu m@10mS$

Exercise 4 - A





- Assuming $gm = 10mS$ from the previous exercise
  - $WI: V_{OV} = -0.05V \rightarrow \frac{gm}{I_{DS}} = 31 \rightarrow I_{DS} = 322\mu A$
  - $MI: V_{OV} = 0.1V \rightarrow \frac{gm}{I_{DS}} = 18 \rightarrow I_{DS} = 555\mu A$
  - $SI: V_{OV} = 0.5V \rightarrow \frac{gm}{I_{DS}} = 4 \rightarrow I_{DS} = 2500\mu A$

Exercise 4 - B

- Gain $A_V = gm \times R_L$
- Voltage at Drain $V_{DS} = V_{DD} - R_L \times I_{DS}$
- Assuming an overdrive voltage of 0.1V
  - $V_{DS} = 1.0V \rightarrow R_L = \frac{1.1V-1.0V}{555\mu A} = 180\Omega \rightarrow \mathrm{A_V} = 1.8\frac{\mathrm{V}}{\mathrm{V}}$
  - $V_{DS} = 0.55V \rightarrow R_L = \frac{1.1V-0.55V}{555\mu A} = 990\Omega \rightarrow \mathrm{A_V} = 9.9\frac{\mathrm{V}}{\mathrm{V}}$
  - $V_{DS} = 0.2V \rightarrow R_L = \frac{1.1V-0.2V}{555\mu A} = 1.62\mathrm{k}\Omega \rightarrow \mathrm{A_V} = 16.2\frac{\mathrm{V}}{\mathrm{V}}$
  - General case: $A_V = gm \times R_L = gm \times \frac{V_{DD}-V_{DS}}{I_{DS}} = \frac{gm}{I_{DS}} \times (V_{DD} - V_{DS})$, so the smaller the drain voltage the higher the gain, but at what cost?

Comparison table

Implementing the same gm=10mS	Width α Area	IDS α Power	Voltage gain gmRL @ VDS = 0.55V
Weak inversion (vov = -0.05)	10000um	370uA	17.0 V/V (24.6dB)
Moderate inversion (vov = 0.1)	500um	714uA	9.9 V/V (19.9dB)
Strong inversion (vov = 0.5V)	50um	1100uA	2.2 V/V (6.8dB)

Plotting dependency to Channel length

Extracting small signal parameters: session1_part2.m

```
%% Circuit
disp('     spec.VDD/2       ');
disp('        |             ');
disp('        |             ');
disp('  Vg---Mnx            ');
disp('        |             ');
disp('       Vs             ');

fprintf('\n--- First Exercise: Designing transistor from scratch ---\n');
```

- Fixed parameters: VDS, W/L
- Sweep parameters: L, VGS
- Calculated parameters: gm, gds, gm/ID (efficiency), gm/gds (self gain)

Think about it!

- Extract the cutoff frequency (Mn.ft) value and plot alongside the previous graphs
  - For higher ft which length and inversion region are preferable?

Resistive load common source stage

```
%% EX1: Circuit
disp('       VDD           ');
disp('        |            ');
disp('        RL           ');
disp('        |----+-OUT   ');
disp('  IN---Mn1   |       ');
disp('        |    CL      ');
disp('        |    |       ');
disp('       GND  GND      ');
```

- Specification: Gain (>18dB), GBW (>100MHz) and Load (1pF) (given by system level decisions)
- Design choices: L, inversion region (VOV = VGS-VTH)
- Useful equations: $gm = 2\pi \times GBW \times C_{load}$ (note that gm turns to be a specification), $A_v = \frac{g_m}{g_{ds}+1/R_L}$

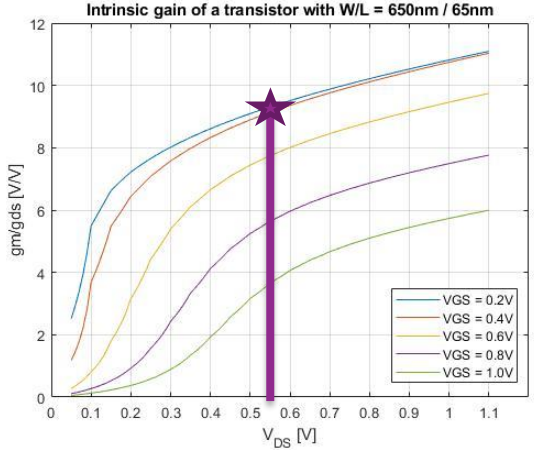
Writing the matlab code

1. GBW = 100MHz, CL = 1pF, Gain > 18dB, Width < 1mm
2. Calculate target gm from CL and GBW
3. Set your design choices (Mn.lg, Mn.vov)
4. Set the terminal voltages (Mn.vds, Mn.vsb, VD = VDD/2)
5. Extract VTH
6. Set gate voltage (Mn.vgs = VOV+VTH)
7. Calculate width to realize given gm (Mn.w = mosWidth('gm', spec.gm, Mn))
8. Extract IDS from mosOpValues
9. Calculate RL from IDS and VD
10. Calculate gain (equation in previous slide)
    - If the specification was met the design is ready to go to Cadence, if not go back to your design choices (lg and VOV) and start again reasoning towards which direction you want to go (long x short channel; weak x strong inversion)

Building some intuition

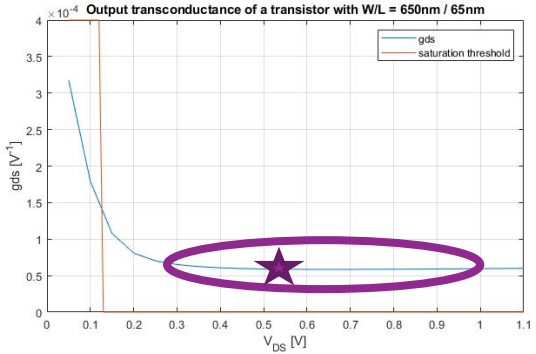
- Exercise 5:
  - ETA: 15 minutes

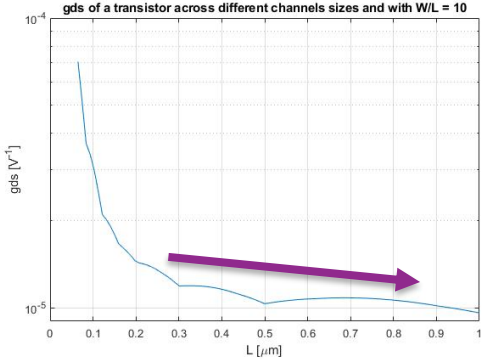
Exercise 5



$$A_V = \frac{gm_{NMOS}}{gds_{NMOS} + gds_{PMOS}}$$

$\frac{gm_{NMOS}}{gds_{NMOS}}$ maximization assuming the PMOS was already minimized





gds_{PMOS} minimization

PMOS load common source stage

```
%% EX2: Circuit
disp('       VDD           ');
disp('        |            ');
disp('       Mp2           ');
disp('        |----+-OUT    ');
disp('  IN---Mn2   |        ');
disp('        |    CL       ');
disp('        |    |        ');
disp('       GND  GND       ');
```

- Specification: Gain (>26dB), GBW (>100MHz) and Load (1pF)
- Gain definition: $A_v = \frac{g_m^N}{g_{ds}^N + g_{ds}^P}$

Writing the Matlab code

1. GBW = 100MHz, CL = 1pF, **Gain > 26dB**, Width < 1mm
2. NMOS design has the same process as the case before
3. Sizing the PMOS load has some hard constraints
    1. Mp.vds + Mn.vds = spec.vdd (forced by supply)
    2. Mp.ids = Mn.ids (to provide a load functionality)
4. PMOS width is calculated based on its required IDS (Mp.ids = mosWidth('ids', Mp.ids, Mp))

Testing the results in LTSPICE

homework

```
fprintf('\n--- Homework 1: Common source amplifier with cascodes ---\n');
%% HW1: Circuit
disp('       VDD           ');
disp('        |            ');
disp('    IN--Mp3          ');
disp('        |            ');
disp('       Mp4           ');
disp('        |            ');
disp('        +----+-OUT    ');
disp('        |    |        ');
disp('       Mn4   |        ');
disp('        |   CL        ');
disp('       Mn3   |        ');
disp('        |    |        ');
disp('       GND  GND       ');
```

```
%% HW1: Specs
spec.fGBW       = 80e6;    % [Hz] GBW frequency
spec.Cl         = 15e-12;   % [F] load capacitance
spec.VDD        = 1.1;      % [V] Power supply voltage
spec.Vswing     = 0.2;      % [V] minimum peak to peak voltage swing of output
bodyeffecton    = 1;
```

- Gain > 40dB
- Bodyeffecton triggers the body effect during biasing and calculations
  - This affects mainly the variation of Vth with respect to the Vsb voltage
  - Long channel devices suffer less from this effect than short channel devices

embracing a better life